Open Access

Christoph J. Salomon*, Sebastian Ley and Marko Helbig

Hardware dependencies of GPU-accelerated beamformer performances for microwave breast cancer detection

DOI 10.1515/cdbme-2016-0111

Abstract: UWB microwave imaging has proven to be a promising technique for early-stage breast cancer detection. The extensive image reconstruction time can be accelerated by parallelizing the execution of the underlying beamforming algorithms. However, the efficiency of the parallelization will most likely depend on the grade of parallelism of the imaging algorithm and of the utilized hardware. This paper investigates the dependencies of two different beamforming algorithms on multiple hardware specification of several graphics boards. The parallel implementation is realized by using NVIDIA's CUDA. Three conclusions are drawn about the behavior of the parallel implementation and how to efficiently use the accessible hardware.

Keywords: beamforming; breast cancer detection; CUDA; GPGPU; GPU; microwave imaging; NVIDIA; UWB.

1 Introduction

Ultra-wideband (UWB) microwave imaging is a promising method for early-stage breast cancer detection, which has been successfully tested by numerous research groups [1–4]. It is based on the differing dielectric properties of tumors and surrounding tissue, originating from their different water content.

For imaging purposes, UWB electromagnetic stimuli are sent into the material under test. Beamforming algorithms are then applied to calculate the reflected energy for numerous focal points inside the area of interest. The tumor detection requires an adequate resolution of the scanning grid, to ensure that all necessary dielectric transitions inside the breast are detected. Depending on the desired image resolution, the number of channels and the complexity of the applied beamformer, respectively, the time for a 3D-Image reconstruction will elevate rapidly.

One approach to accelerate these computations is to shift the imaging process from serial execution on a Central Processing Unit (CPU, also referred to as host) to massively parallel execution on a Graphics Processing Unit (GPU, also referred to as device). This paradigm is known as General Purpose Programming of Graphics Processing Units (GPGPU). Utilizing the many-core architecture of GPUs, several stages of the imaging process can be computed simultaneously by different computational cores, resulting in a massive reduction of the image reconstruction time [5]. However, various types of beamformers are available for breast cancer detection, all differing in characteristics and complexity as described in [1]. To make efficient use of GPGPU, the accelerating hardware should match the achievable parallelism of the applied algorithm.

This paper presents a parallel solution for a Delay-and-sum (DAS) beamformer and a Delay-multiply-and-sum (DMAS) beamformer, respectively, and the acceleration achieved for different types of graphics boards. The boards vary in their GPU's processor architecture (Maxwell and Kepler), the amount of their GPU's processor cores (CUDA-Cores), their clock rate and their memory bandwidth, respectively. The runtime of the parallel execution is presented as a function of pixel quantity and benchmarked against a fully serial implementation computed by a general purpose CPU. To utilize the GPU, NVIDIA's compute unified device architecture (CUDA) provides the necessary programming interface. In the following sections we describe the applied beamforming algorithms with respect to their potential parallelism, our way of implementation and the specifications that may have influence on the overall performance. We present the achieved acceleration on different hardware architectures and draw a conclusion about factors that have to be considered for an efficient use of parallel hardware when accelerating beamforming algorithms for medical UWB imaging.

***Corresponding author: Christoph J. Salomon,** Biosignal Processing Group, Technische Universität Ilmenau, E-mail: christoph.salomon@tu-ilmenau.de
Sebastian Ley and Marko Helbig: Biosignal Processing Group, Technische Universität Ilmenau, E-mail: sebastian.ley@tu-ilmenau.de (S. Ley); marko.helbig@tu-ilmenau.de (M. Helbig)

2 Imaging algorithm

The DAS algorithm is the most basic implementation of a beamformer. Based on [4] it can mathematically be described as

$$I(r_0) = \left[\sum_{n=1}^{N} S_n\Big(\tau_n(r_0)\Big) \right]^2 . \quad (1)$$

$I(r_0)$ is the backscattered energy of the focal point r_0, with $r_0 = [x_0, y_0, z_0]$. $S_n(t)$ represents the clutter subtracted signal of channel n, τ_n the corresponding signal runtime along the path Tx_n-r_0-Rx_n (Tx_n: transmitting antenna of channel n; Rx_n: receiving antenna of channel n) and N the total number of channels. To calculate the reflected energy of an arbitrary point inside the area of interest, the Euclidean distance between the antennas and the coordinates of the focal point has to be calculated. Considering signal propagation time and sampling rate, the representing sample of the respective channel for this point is identified. To increase the signal-to-noise ratio (SNR), these steps are repeated for numerous channels and the results are summed up. Iterating through a matrix of focal points and color coding the results creates a visually interpretable image of reflected signal energy inside the breast.

To further increase the SNR, Lim et al. [6] introduced the DMAS algorithm which contains an additional pairwise multiplication of all channels. Deriving from their presented diagram the DMAS can mathematically be described as

$$I(r_0) = \left[\sum_{n=1}^{N-1} \sum_{m=n+1}^{N} S_n\Big(\tau_n(r_0)\Big) S_m\Big(\tau_m(r_0)\Big) \right]^2 . \quad (2)$$

By examining equations (1) and (2) it becomes obvious, that the reflected energy for all focal points can be computed independently. However, the calculation itself contains serial segments that cannot be parallelized. To sum all intermediate results, the same memory space has to be addressed at one point of the calculation for each channel, which limits the achievable parallelism of the overall calculation. The additional pairwise multiplication and addition inside the DMAS algorithm restricts this parallelism even more. The ratio of serial and parallel executable parts differs for both algorithms. Therefore, the efficiency of the parallel solution for the respective beamforming algorithm will most likely depend on the grade of parallelism of the utilized hardware.

3 Parallel implementation

The following section describes the specifications and settings of our CUDA implementation that may have influence on the overall performance of the parallel execution, including data transfer and hardware initialization.

Since host and device memory are physically separated, all required data has to be transferred to the device memory first, before it can be processed by the GPU. The backscattered signals and coordinates of antennas and focal points are copied to the device before the first image point is computed. Results are transferred back to the host memory only after all computations of the graphics card have been completed. Due to intelligent caching behavior, no memories of the GPU except for global memory (DRAM) are used. All data is kept in page-locked memory on the host before being copied. Once the data is available to the device, the parallel computation is started.

To utilize graphics boards for general-purpose computations, CUDA maps the physical computation cores of the GPU to a three-dimensional grid of threads. Threads are then again organized in blocks. The number of threads per block has direct impact on the processor occupancy and the achieved acceleration. A block-size of $8 \times 8 \times 1$ has proven to be most efficient in average for multiple graphics cards and is used for all our computations. As the number of computed image points varies, the grid-size is adapted in a way that every thread computes one single image point. Both beamformers are implemented as a function that is executed by every thread, the so-called kernel. The kernel processes all necessary steps described in (1) and (2), respectively.

4 Accelerated püerformance

To illustrate the dependencies of reconstruction time as a function of pixel quantity, different amounts of image layers are reconstructed during a single program execution. Although the scanning matrix can be chosen arbitrarily, for the purpose of comparison every reconstructed image layer lies perpendicular to the axis of rotational symmetry of the breast phantom and consists of 10201 pixels. This matches a resolution of 1 mm per plane for an image dimension of 10 cm $\times$ 10 cm. Furthermore, every image point is computed applying 4096 back scattered signals resulting from 128 channels as described in [2].

A benchmark is set by first executing a complete serial implementation of both beamformers on a general purpose CPU of type Intel Core i5 330. The results of this

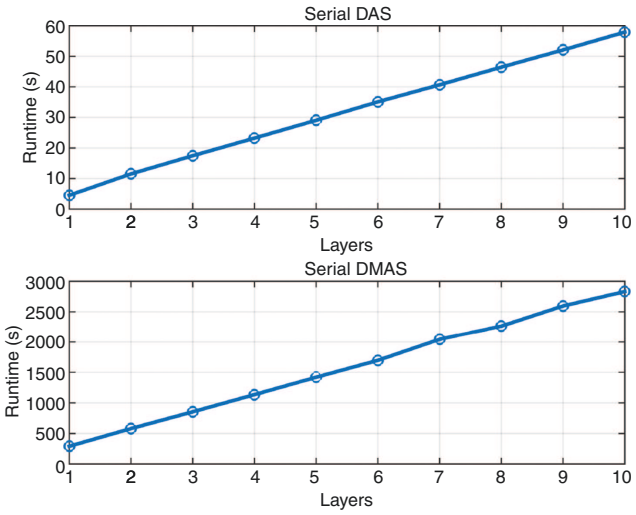

Figure 1: Execution time of serial DAS (top) and DMAS (bottom), respectively on a single computation core of a CPU type Intel Core i5 330 at 3.33 GHz. Runtime is presented as a function of reconstructed image layers. Each layer consists of 10201 pixels, corresponding to 1 mm image resolution for an image dimension of 10 cm × 10 cm.

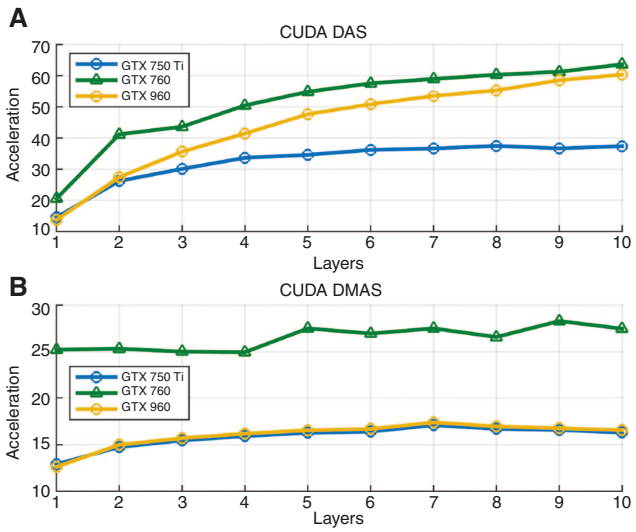

Figure 2: Achieved acceleration of parallel beamformer execution. Top (A): DAS Beamformer. Bottom (B): DMAS Beamformer. Acceleration refers to the results from Figure 1 and is the ratio of serial execution time and parallel execution time. Triangles represent GPUs with Kepler architecture; Dots refer to GPUs with Maxwell architecture.

execution are presented in Figure 1. The image reconstruction is executed on a single computation core at 3.33 GHz clock rate. Recordings of runtime are implemented in the serial code using time stamp functions of the C standard-libraries.

Since the kernel runs asynchronous to the host code, casual timestamps are not applicable to determine the parallel execution time. For that purpose CUDA events were used. The achieved acceleration is illustrated in Figure 2 for different graphics boards. Acceleration refers to the results presented in Figure 1 and is the ratio of serial execution time and parallel execution time, including the time necessary for copying data between host and device. During the execution we tried to minimize the number of running processes and interrupts. The exact hardware specifications of all utilized graphics boards can be seen in Table 1.

Figure 3 illustrates a reconstructed image layer using a DAS and a DMAS algorithm, respectively. A tumor imitation is placed inside the breast phantom as described in [2].

Table 1: Hardware specifications of utilized graphics boards.

	GTX 750 Ti	GTX 760	GTX 960
CUDA-Cores	640	1152	1024
Clock (MHz)	1024	980	1127
Memory-bandwidth (GB/s)	86.4	192	112
GPU architecture	Maxwell	Kepler	Maxwell
DRAM (MB)	2048	2048	2048

5 Conclusion

Evaluating the results, we can draw three conclusions about how to use the presented hardware efficiently for GPU-accelerated microwave breast imaging.

First: The computation shifted to the GPU has to be as extensive as possible. Figure 2(A) shows, that with increasing problem size more acceleration is achieved until a certain point of saturation. For an effective use the hardware has to operate at full capacity. Due to CUDA's intelligent thread-management this capacity cap is not the amount of computation cores but has to be determined empirically.

Second: The implementation performance of a highly parallel algorithm is mostly dependent on the number of processing cores available on the accelerating hardware. Apparently, the highest acceleration is achieved with the GTX 760, which offers the lowest clock rate but the highest number of CUDA-Cores. Therefore, clock rate is the least significant factor for an effective choice of hardware. However, the GTX 760 also offers the highest memory bandwidth. Examining Figure 2(A) it appears that with higher problem size, the acceleration achieved with the GTX 970 approaches the GTX 760's performance. That is, even the GTX 760 features a much higher memory bandwidth. The gap between the amounts of computational

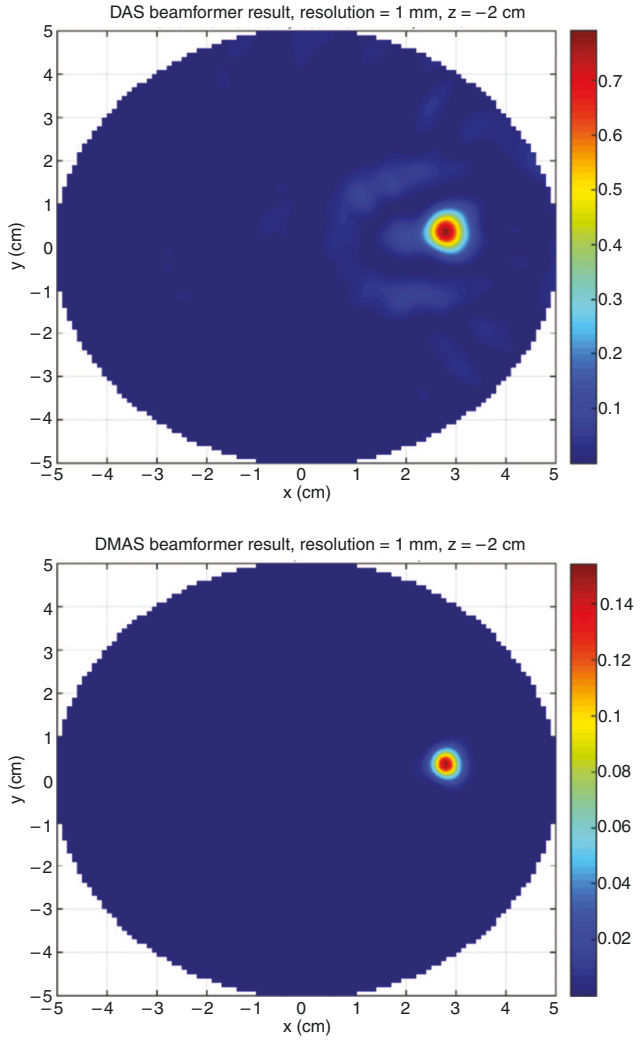

Figure 3: Reconstructed image layer of breast phantom with a tumor imitate of 15 mm diameter. Resolution is 1 mm. Top: DAS-algorithm applied. Bottom: DMAS-algorithm applied. Linear scale of energy values.

cores however, is a lot smaller. We conclude, that the number of computational cores has a higher impact on the performance, than the memory bandwidth. When executing a less parallel problem, the decisive component changes, which leads to our next conclusion.

Third: An accelerated problem that still contains a lot of serial computations after the parallelisation is more efficiently executed by NVIDIA GPUs with Kepler architecture than by GPUs with Maxwell architecture. When applying the DMAS algorithm, again the GTX 760 scores the best results. GTX 960 and GTX 750 show nearly equal results, even though their hardware specifications differ a lot. However, both GPUs have the same hardware architecture. The main difference between the graphics boards disregarding number of computation cores, memory bandwidth and clock is the GPU architecture.

All presented imaging algorithms imply independency of the computed image points among each other. However, some algorithms include certain dependencies among neighboring focal points. Future investigations have to focus on efficient ways of accelerating these methods, where our approach of parallelisation is not applicable. Additionally, a broader study of graphics cards will probably lead to a more general conclusion about efficient use of accelerating hardware.

Author's Statement

Conflict of interest: Authors state no conflict of interest. Material and Methods: Informed consent: Informed consent has been obtained from all individuals included in this study. Ethical approval: The research related to human use complies with all the relevant national regulations, institutional policies and was performed in accordance with the tenets of the Helsinki Declaration, and has been approved by the authors' institutional review board or equivalent committee.

References

[1] Byrne D, O'Halloran M, Glavin M, Jones E. Data independent radar beamforming algorithms for breast cancer detection. Prog Electromagn Res. 2010;107:331–48.
[2] Helbig M, Dahlke K, Hilger I, Kmec M, Sachs J. Design and test of an imaging system for UWB breast cancer detection. Frequenz. 2012;66:387–94.
[3] Klemm M, Leendertz JA, Gibbins D, Craddock IJ, Preece A, Benjamin R. Microwave radar-based differential breast cancer imaging: imaging in homogeneous breast phantoms and low contrast scenarios. IEEE Trans Antennas Propag. 2010;58:2337–44.
[4] Li X, Davis SK, Hagness SC, Van der Weide DW, Van Veen BD. Microwave imaging via space-time beamforming: experimental investigation of tumor detection in multilayer breast phantoms. IEEE Trans Microw Theory Tech. 2004;52:1856–65.
[5] Elahi MA, Shahzad A, Glavin M, Jones E, O'Halloran M. GPU accelerated confocal microwave imaging algorithms for breast cancer detection. The 9th European IEEE Conference on Antennas and Propagation (EuCAP) 2015:1–2.
[6] Lim HB, Nhung NTT, Li EP, Thang ND. Confocal microwave imaging for breast cancer detection: delay-multiply-and-sum image reconstruction algorithm. IEEE Trans Biomed Eng. 2008;55:1697–704.